IN THE SPECIFICATION:

Please replace the paragraph beginning on page 4, line 4 with the following rewritten paragraph:

--The problem with texture mapping systems that accomplish the texture addressing and look-up in a brute force method, is that the transfer of data between the various processes require wide memory buses, multiple and/or multi-ported memory subsystems, and/or multiple clock cycles to transfer the required data. Some of these systems also are penalized with the time required to load the texture maps or large chunks of the maps into a specialized memory system prior to rendering of primitives that use it. Texture mapping has been accomplished with special purpose memory devices or a special purpose memory system that can offer effective random access anywhere within a texture map.--

Please replace the paragraph beginning on page 16, line 6 with the following rewritten paragraph:

--Referring now to Figure 2, there is shown a block diagram of the three-dimensional texture caching system of the present invention. The cache system 24 includes a cache controller 26, a cache arbiter 28 and a texture cache memory 30. Also shown in Figure 2 are a texture address calculator 32, a texture main memory 34 and an interpolator 36. The texture address calculator 32 calculates the texture memory addresses (U,V) for each pixel and also the specific LODs from which the texture addresses are to be retrieved. The cache controller 26 determines if the cache memory 30 will contain the texture data for a given address or if it needs to be fetched. It allocates space for new data to be fetched and determines the location of the data required for each pixel to pass tag data to the cache arbiter 28, which will be used to schedule data entry and access from the cache memory 30. The cache controller 26 also determines the order in which data will be overwritten. Based on this determination, the cache controller 26 fetches data from the texture main memory 34 and transfers it into the appropriate addresses in the texture cache memory 30 when that cache arbiter 28 determines the data located in that location is no longer in use. In accordance with a preferred embodiment

of the present invention, the cache controller 26 is comprised of four individual controllers 38, 40, 42 and 44, each of which corresponds to the partitioning of data into four different data groups in the texture main memory 34. This partitioning of data also applies to the four memory banks in the texture cache memory 30. The controllers 38,40,42,44 regulate and keep track of what is stored in the memory banks of the cache memory 30. In the preferred embodiment, each of the controllers contains four stages with each stage referencing a double quad word in the respective cache memory bank. The memory banks are organized into an array of W, X, Y and Z texel blocks, partitioned by A, B, C, and D texels in each array. The cache arbiter 28 determines if it can write the next data values into the cache without overwriting any data that is still needed. Once the appropriate texture data is determined to be present in the cache memory 30, the controller outputs the appropriate texel data into the interpolator 36 to perform the required interpolation such as bilinear interpolation.--

A2 CONT.

Please replace the paragraph beginning on page 17, line 23 with the following rewritten paragraph:

--Figure 3 shows a texture map that is to be stored in main memory 46 being partitioned by labeling or identifying each texel with four different identifiers. As shown, texels are labeled A, B, A, B etc. across every even row of the map and likewise C, D, C, D etc. across every odd row of the map. This organization results in an interesting relationship when bilinear interpolation is performed. After mapping a pixel into the texture map, the texture addresses to the texels that surround the mapped pixel result in a group of texels, no matter what four are selected, that will always be consisting on one A, one B, one C, one D type of texel. This allows the organization of texel types in cache memory to be in banks so that all four texels that are needed for a pixel can be accessed in one clock cycle. This is accomplished by putting all A types in one bank, all B types in another bank, C types in a third bank, and D types in a fourth bank. The mapping of pixel 48 into texture map 46 is an example showing that one of each texel type will be selected for bilinear interpolation.--

A3

Please replace the paragraph beginning on page 19, line 19 with the following rewritten paragraph:

--In accordance with the present invention, an optimal texture block size can be determined to simplify caching. From a topological point of view and from the random nature of the alignment of texels and pixels, the following relationship should be satisfied in order to maintain good texture coherency. First, the texel block arrangement should be compact such as a square or at most a two to one aspect ratio rectangle. Second, the texel block size should be compatible or multiples with the texture data transfer size per clock. Third, the texture main memory array should be organized so that it can be retrieved one block at a time from a single page of memory to allow the use of inexpensive memory devices. For example, if the bus width for transfer of data from main memory to cache memory is 128 bits wide per clock cycle, the number of transfers per request and the texel depth (size in bits per single texel) determines the organization. If texels are 8, 16 or 32 bits and the number of transfers per request is 1 then the optimal block sizes are 4x4, 2x4, and 2x2 respectively. The texture main memory array can be made from inexpensive memory devices because the data is organized for continuous block transfer per clock cycle.--

A4

Please replace the paragraph beginning on page 21, line 18 with the following rewritten paragraph:

--As shown in Figure 5, when texture data is moved from texture main memory to texture cache memory, the texels 50 undergo a conversion or reorganization. They are stored in the texture main memory array so that a double quad word contained in texel block 52 can be accessed and sent across the bus 64. The texture cache memory storage requires a reorganization of the elements so that they are stored in a form necessary for access per clock cycle in the bilinear texture interpolator. This is achieved by storing the texels of the double quad word in four cache memory banks 56,58,60,62. A double quad word is made up of an even and an odd part. The A texels of double quad word 52 are stored in bank A, the B texels are stored in bank B, the C texels are stored in bank C and the D texels are stored in bank D.--

A5